

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4).

Dated: January 14, 2008

Electronic Signature for Michael G. Verga: /Michael G. Verga/

PATENT

**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In the Matter of the

Application of: REES, *et al.*

SERIAL NO.: 10/640,625

Filed: August 12, 2003

Entitled:

**PROPAGATING WEB  
TRANSACTION CONTEXT INTO  
COMMON OBJECT MODEL (COM)  
BUSINESS LOGIC COMPONENTS**

Docket No.:

10017133-1

Group Art Unit: 2194

Examiner: PANTOLIANO, Jr.,  
Richard

Mail Stop: Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**APPEAL BRIEF PURSUANT TO 37 C.F.R. § 41.37**

TABLE OF CONTENTS

I.	REAL PARTY IN INTEREST.....	1
II.	RELATED APPEALS AND INTERFERENCES.....	1
III.	STATUS OF CLAIMS.....	1
IV.	STATUS OF AMENDMENTS.....	1
V.	SUMMARY OF CLAIMED SUBJECT MATTER.....	1
VI.	GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL.....	6
VII.	ARGUMENT.....	6
	CLAIMS APPENDIX.....	15
	EVIDENCE APPENDIX.....	19
	RELATED PROCEEDINGS APPENDIX.....	20

## I. REAL PARTY IN INTEREST

The real party in interest is Hewlett-Packard Development Company, L.P. of Houston, Texas. Hewlett-Packard Development Company, L.P. derives its rights in this application by virtue of assignment of the application to Hewlett-Packard Development Company, L.P.

## II. RELATED APPEALS AND INTERFERENCES

None

## III. STATUS OF CLAIMS

Claims 1-12 and 15-23 are currently pending in the present application, Application Number 10/640,625. Claims 13 and 14 were previously cancelled. Claims 1-12 and 15-23 have been finally rejected and, therefore, are subject to appeal.

## IV. STATUS OF AMENDMENTS

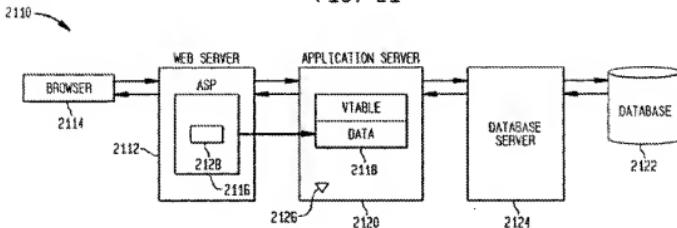
All Amendments have been entered.

## V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 (pg. 47, lns 4-18) is directed to a method of instrumenting a Common Object Model (COM) object 2118 invoked by a client for performing a selected business logic. This method comprises intercepting a request from the client for creating said COM object 2118. (*See e.g.*, Specification, pg. 38, lns. 15-16; pg. 39, lns. 20-22.) The method additionally comprises generating a wrapper object 2132 corresponding to said requested COM object 2118. (*See e.g.*, Specification, pg. 38, lns. 17-19; pg. 39 lns. 20-22;

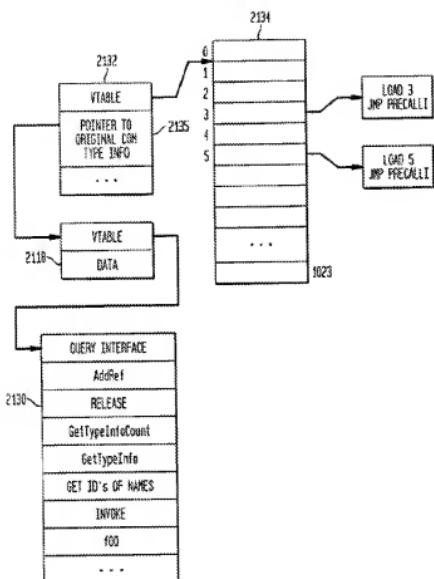
and wrapper object 2132 and COM object 2118 of FIGs. 21, which is reproduced below.)

*FIG. 21*



The wrapper object 2132 implements a universal interface 2134 having a plurality of virtual functions 1024 each indexed by a number corresponding to an index number of a method associated with an interface 2134 of said requested COM object 2118 providing said client with a reference pointer 2136 to said wrapper COM object 2132. (See e.g., Specification pg. 38, lns. 22-27; wrapper object 2132 and universal interface 2134 of FIG. 23, which is reproduced below.)

FIG. 23



Additionally, pursuant to this method, upon invocation of a method associated with an interface of the requested COM object 2118 by the client, a virtual function 1024 is invoked of the universal interface 2134 of the wrapper object 2132. (See e.g., Specification, pg. 39, ln. 30 – pg. 40 ln. 2.) This universal interface 2134 is indexed by a number corresponding to an index number of said requested method. (See e.g., Specification, pg. 39, ln. 30 – pg. 31 ln. 2; also see e.g., pg. 40, lns. 19-22.) The invoked virtual function references instructions for saving a start time marker, executing instructions corresponding to said requested method, and saving a stop time marker upon completion of execution of said instructions associated with the requested method. (See e.g., Specification, pg. 40 ln. 30 – pg. 41 ln. 8; and FIG. 24.)

Independent claim 20 (pg 49, lns 23-30) is directed to a system for monitoring a response time of a transaction performed by one or more Common Object Model (COM) objects 2118 executing on one or more platforms 2146 and 2148. (*See e.g.*, Specification, FIG. 21) This system comprises one or more monitoring agents deployed on selected ones of said platforms 2146 and 2148. (*See, e.g.*, monitoring agent 2126; pg. 38, lns. 11-15.) Each monitoring agent can intercept a request for creating at least one of said COM objects 2118 and generate a wrapper object 2132 corresponding to said requested COM object 2118. (*See e.g.*, Specification, pg. 38, lns. 15-19; pg. 39 lns. 20-22; and wrapper object 2132 of FIG. 21 and COM object 2118 of FIGS. 21 and 23.) The wrapper object 2132 implements a universal interface 2134 having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object 2118. (*See, e.g.*, Specification, pg. 38, lns. 22-27, wrapper object 2132, universal interface 2134 of FIG. 23.)

Independent claim 23 (pg. 50, lns 8-14) is directed to a computer readable medium storing instructions for performing a method of instrumenting a Common Object Model (COM) 2118 object invoked by a client comprising intercepting a request from the client for creating said COM object 2118. (*See e.g.*, Specification, pg. 38, lns. 15-16; pg. 39, lns. 20-22.) The method further comprises generating a wrapper object 2132 corresponding to said requested COM object 2118. (*See e.g.*, Specification, pg. 38, lns. 17-19; pg. 39 lns. 20-22; and wrapper object 2132 of FIG. 21 and COM object 2118 of FIGS. 21 and 23.) The wrapper object 2132 implements a universal interface 2134 having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object 2118. (*See e.g.*, Specification, pg. 39, ln. 30 – pg. 31 ln. 2; *also see e.g.*, pg. 40, lns. 19-22.) The method additionally comprises providing said client with a reference pointer to said wrapper COM object 2118. (*See e.g.*,

Specification pg. 38, lns. 21-22.)

## **VI.GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL**

1. Whether the Examiner improperly rejected independent claims 20 and 23 under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 6,263,491 to Hunt (hereinafter “Cobb”) when Cobb fails to disclose a wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object, as recited by claim 20 and 23.
2. Whether the Examiner improperly rejected independent claim 1 under 35 U.S.C. § 103(a) as unpatentable over Hunt in view of U.S. Patent No. 6,041,352 to Burdick (hereinafter “Burdick”) when neither Hunt nor Burdick teach or suggest invoking a virtual function of said universal interface of the wrapper object indexed by a number corresponding to an index number of said requested method, as recited by claim 1.

## **VII.ARGUMENT**

The following arguments address the above groups of claims based on the similarity of the rejections levied by the Examiner and/or by the similarity of the Applicants’ basis for traversing such rejections.

**A. Hunt fails to disclose a wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object, as recited by claim 20 and 23**

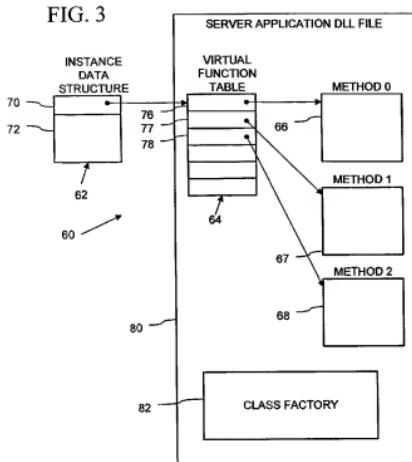
In the Office Action mailed March 20, 2007, the Examiner rejected independent claims 20 and 23 under 35 U.S.C. § 102(b) as being anticipated by Hunt. For at least the reasons provided below, Hunt fails to teach every element of claims 20 and 23, and, as such,

the rejection should be reversed.

An anticipating “reference must describe and enable the claimed invention, including all claim limitations, with sufficient clarity and detail to establish that the subject matter already existed in the prior art and that its existence was recognized by persons of ordinary skill in the field of the invention.” (*See, Crown Operations Int'l, Ltd. v. Solutia Inc.*, 289 F.3d 1367, 1375, 62 USPQ2d (BNA) 1917, 1921 (Fed. Cir. 2002).) The Examiner has the burden of establishing a *prima facie* case of anticipation by pointing out where each and every element of the claimed invention is described in a single prior art reference in a manner sufficient to have placed a person of ordinary skill in the art in possession thereof. (*See, In re Spada*, 911 F.2d 705, 708, 15 USPQ2d (BNA) 1655, 1657 (Fed. Cir. 1990).)

Hunt discloses a COIGN system that identifies Component Object Model (COM) components, measures communications between COM components, classifies COM components, measures network behavior, detects component location constraints, generates optimal distribution schemes, and distributes COM components during runtime. (*See, Hunt*, col. 5, lns. 24-30.) COIGN is described as a component based application that is developed as a package of component objects that conform to Microsoft's COM specification. (*See, Hunt*, col.7 ln. 66- col. 8 ln. 5.) FIG. 3 of Hunt, which is reproduced below, illustrates a Microsoft COM object 60 for the COIGN system of Hunt that is executed by a computer. (*See, Hunt*, col. 7 ln. 66 – col. 8 ln. 5).

FIG. 3

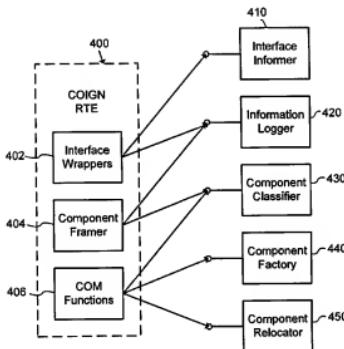


As illustrated, this COM object 60 includes an instance data structure 62, a virtual function table 64, and methods 66-68. (See, Hunt, col. 8 lns. 30-34.) The instance data structure 62 contains a pointer 70 to the virtual function table 64 and data 72. (See, Hunt, col. 8 lns. 34-36.) The virtual function table 64 contains entries 76-78 each of which contains a reference to a corresponding member method 66-68. (See, Hunt, col. 8 lns. 38-41.)

Objects and programs (referred to as "clients" of the COM object 60) are disclosed as accessing the functionality of the COM object 60 by invoking its member methods 66-68 through a COM object interface. (See, Hunt, col. 9 lns. 48-51.) Before the member methods 66-68 may be accessed by a particular client, the COM object must first be instantiated (*i.e.*, by causing the class factory to create the instance data structure 62 of the object), and the client must be provided with an interface pointer to the COM object. (See, Hunt, col. 9 lns. 51-54.)

FIG. 13, which is reproduced below, illustrates an architecture of the COIGN system of Hunt. (See, Hunt, col. 37 lns. 33-36.)

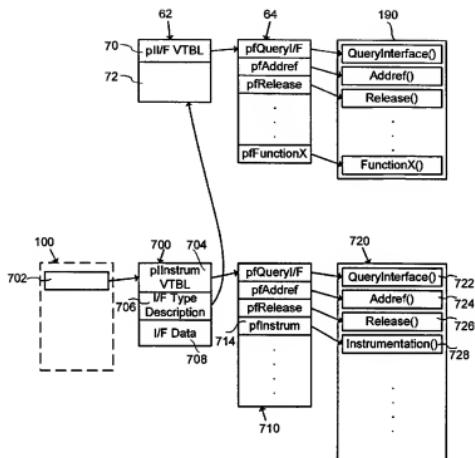
FIG. 13



As illustrated, the COIGN system's architecture includes a run time executive (RTE) 400. Hunt discloses that this RTE "wraps" all COM interfaces by replacing the pointers to the COM interfaces with pointers to a COIGN instrumentation interface wrapper 402. (See, Hunt, col. 38 lns. 23-25.)

FIG. 17 of Hunt, which is reproduced below, illustrates a COM object (e.g., the illustrated instance data structure 62, virtual function table 64, and member function 190), an interface wrapper 700 appended to the COM object, and analytical data 708 appended to the wrapped COM object. (See, Hunt, col. 5 lns. 4-7.)

FIG. 17



As illustrated in FIG. 17, the interface wrapper 700 holds a pointer 704 to a virtual table 710 of the COIGN instrumentation system and a interface type description 706 for the wrapped interface. (See, Hunt, col. 45 ln. 63- col. 46 ln. 1.) The interface type description 706 includes information that can be used to access the component interface through the instance data structure 62 and pointer 70 to the virtual table 64 for the interface. (See, Hunt, col. 46 ins. 1-5.) The virtual table 710 for the COIGN instrumentation system includes pointers 711-713 to unknown functions 722-726m and a pointer 714 to an instrumentation function 728.

When a client 100 attempts to invoke an interface member function 190, the pointer 702 to the interface wrapper 700 is followed and COIGN has intercepted the member-

function invocation. (*See, Hunt, col. 46 lns. 12-15.*) An instrumentation function 728 is invoked that processes member-function parameters and then calls the component interface of the member function using the information supplied in the interface type description 706. (*See, Hunt, col. 46 lns. 15-19.*)

Independent claim 20 recites, in part, “wherein said wrapper object implements a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object.” In rejecting claim 20, the Examiner alleged that Col. 8 lines 22-49 and FIG. 3 of Hunt allegedly disclose implementing a universal interface having a plurality of functions. (*See, Office Action at 3.*) The Examiner further stated that the virtual function table of Hunt allegedly corresponds to the universal interface of claim 20. (*See, Office Action at 2.*)

As noted above, FIG. 3 of Hunt illustrates a COM object 60. As such, the relied on virtual function table of Hunt is disclosed as being included in the COM object itself. In other words, the Examiner relied on the virtual function table of FIG. 3 of Hunt as allegedly disclosing a universal interface; however, FIG. 3 illustrates that the illustrated virtual function table is included in COM object 60. As such, the virtual function table of Hunt relied on by the Examiner is not implemented by a wrapper object. Rather, this virtual function table 64 is implemented by the COM object. Thus, this virtual function table of FIG. 3 of Hunt cannot correspond to the universal interface recited in Applicants’ claim 20, which, as noted above recites, “wherein said wrapper object implements a universal interface.” Applicants therefore respectfully submit that the Examiner’s rejection of claim 20 is improper for at least this reason.

If the Examiner intended not to rely on the virtual table of FIG. 3, but instead the virtual table 710 of Fig. 17, Applicants respectfully submit that such reliance would be misplaced for the following reasons.

In the Remarks section of the outstanding Office Action, the Examiner states “the wrapper is accessed in the exact same manner as the intended COM object would, with index numbers being used to access the virtual function table.” (*See*, Office Action at 13.) No support, however, is provided for the assertion that index numbers are used to access the virtual function table. Rather, the Examiner merely states elsewhere in the Office Action that the functions of the interfaces associated with a COM object are inherently indexed because the standardized binary format of the COM virtual function table allows this. (*See*, Office Action at 4-5.) Even if possible, Hunt fails to disclose that each function of Hunt’s virtual table 710 is indexed by number and that this number corresponds to an index number of member function 190-195 of the COM object 60. Rather, as noted above, Hunt teaches that the virtual table 710 includes a pointer 714 that points to an instrumentation function 728 that is disclosed as being included in a group of functions 720 separate from the member functions 190-195 of the COM object. (*See*, Hunt, FIG. 17.) When a member function is to be used, the instrumentation function 728 processes member-function parameters and calls the component interface of the member function using the information supplied in the interface type description 706.

Thus, this virtual table 710 of Hunt functions in a different manner than the universal interface of the present invention, and does not include virtual functions that are each “indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object,” as required by Applicants’ independent claim 20. Rather, the virtual function table of Hunt includes a pointer to a separate and distinct instrumentation function that then processes received information to call a component interface.

Accordingly, Hunt fails to teach or suggest “wherein said wrapper object implements a universal interface having a plurality of virtual functions each indexed by a number

corresponding to an index number of a method associated with an interface of said requested COM object,” as recited by claim 20. Applicants therefore respectfully request that the rejection to claim 20 be withdrawn for at least this reason.

Similarly, independent claim 23 recites “said wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object.” Applicants therefore respectfully submit that for at least similar reasons to those discussed above, independent claim 23 is likewise in condition for allowance.

**B. Neither Hunt nor Burdick, whether taken alone or in combination, teach or suggest invoking a virtual function of said universal interface of the wrapper object indexed by a number corresponding to an index number of said requested method, as recited by independent claim 1.**

In the Office Action mailed March 20, 2007, the Examiner rejected independent claim I under 35 U.S.C. § 103(a) as being unpatentable over Hunt in view of Burdick. For at least the reasons provided below, neither Hunt nor Budick teach or suggest every element of claim I, and, as such, the rejection should be reversed.

To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). “All words in a claim must be considered in judging the patentability of that claim against the prior art.” *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Independent claim 1 recites, in part, “invoking a virtual function of said universal interface of the wrapper object indexed by a number corresponding to an index number of said requested method.” In rejecting claim 1, the Examiner relied on the bases discussed above with reference to independent claim 20 for alleging that Hunt discloses this limitation of claim 1. (See, Office Action at 5.)

As such, for at least similar reasons to those discussed above, Hunt fails to teach or suggest “invoking a virtual function of said universal interface of the wrapper object indexed by a number corresponding to an index number of said requested method,” as recited by independent claim 1. Burdick does not cure, nor has the Examiner alleged that it cures, this defect of Hunt. Applicants, therefore, respectfully submit that the rejection of independent claim 1 is improper and should be withdrawn.

### C. Conclusion

For the reasons noted above, Applicant submits that the pending claims define patentable subject matter. Accordingly, Applicant request that the Examiner’s rejection of these claims be reversed and that the pending application be passed to issue.

Dated: January 14, 2008

Respectfully submitted,

Electronic signature: /Michael G. Verga/  
Michael G. Verga  
Registration No.: 39,410  
CONNOLLY BOVE LODGE & HUTZ LLP  
1875 Eye Street, N.W.  
Suite 1100  
Washington, DC 20006  
(202) 331-7111 (Tel)  
(202) 293-6229 (Fax)  
Agent for Applicant

## CLAIMS APPENDIX

1. A method of instrumenting a Common Object Model (COM) object invoked by a client for performing a selected business logic, comprising:

intercepting a request from the client for creating said COM object,

generating a wrapper object corresponding to said requested COM object, said wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object,

providing said client with a reference pointer to said wrapper COM object,

upon invocation of a method associated with an interface of the requested COM object by the client, invoking a virtual function of said universal interface of the wrapper object indexed by a number corresponding to an index number of said requested method,

wherein said invoked virtual function references instructions for saving a start time marker, executing instructions corresponding to said requested method, and saving a stop time marker upon completion of execution of said instructions associated with the requested method.

2. The method of claim 1, further comprising registering said invoked method with an Application Response Measurement (ARM) agent upon invocation of said method by the client.

3. The method of claim 2, wherein saving a start time marker comprises invoking said ARM agent for generating a record for a transaction corresponding to invocation of said method of the requested COM object.

4. The method of claim 3, wherein saving a stop time marker comprises invoking the ARM agent to generate said stop time marker.

5. The method of claim 1, wherein said wrapper COM object comprises a data structure for storing a number of arguments and a type of each argument associated with each method of said requested COM object.

6. The method of claim 1, wherein said wrapper object comprises a reference pointer for referring to said requested COM object.

7. The method of claim 1, wherein said wrapper object comprises a reference pointer for referring to said universal interface.

8. The method of claim 1, further comprising

defining a policy that indicates whether to perform said step of generating a wrapper COM object corresponding to said requested COM object; and

determining whether to perform said step of generating a wrapper COM object using said policy.

9. The method of claim 8, wherein said policy identifies a requested proxy object, a requested COM object belonging to an MTS package, and a requested COM<sup>+</sup> object for wrapping.

10. The method of claim 8, further comprising

storing said policy in a tabular format in a registry of a system on which said COM objects are executed.

11. The method of claim 1, wherein the step of intercepting a request comprises

patching code associated with one or more selected system functions.

12. The method of claim 11, wherein said system functions are provided in a dynamic link library.

13-14. (Canceled)

15. The method of claim 11, further comprising

for each of said selected system functions, utilizing a hook associated therewith to refer to a program for patching said system function.

16. The method of claim 15, wherein said hook comprises a designated string stored in system registry.
17. The method of claim 16, wherein said patching of the system function comprises replacing selected bytes in a code corresponding to said system function with a jump instruction to a code for creating said requested COM object and generating said wrapper object.
18. The method of claim 17, further comprising copying instructions in said system function code corrupted by said inserted jump instruction to an allocated data area.
19. The method of claim 18, further comprising decoding said selected bytes prior to their replacement by said jump instruction to determine a total number of bytes corresponding to instructions corrupted by said inserted jump instruction.
20. A system for monitoring a response time of a transaction performed by one or more Common Object Model (COM) objects executing on one or more platforms, comprising one or more monitoring agents deployed on selected ones of said platforms, each monitoring agent intercepting a request for creating at least one of said COM objects and generating a wrapper object corresponding to said requested COM object, wherein said wrapper object implements a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object.
21. The system of claim 20, further comprising an Application Response Measurement (ARM) agent in communication with said monitoring agent.

22. The system of claim 21, wherein each of said virtual functions refers to a code having instructions for invoking said ARM agent prior to execution of an invoked method of a wrapped COM object to save a start time marker, executing said invoked method, and invoking said ARM agent subsequent to completion of execution of said invoked method to save a stop time marker.
23. A computer readable medium storing instructions for performing a method of instrumenting a Common Object Model (COM) object invoked by a client comprising intercepting a request from the client for creating said COM object, generating a wrapper object corresponding to said requested COM object, said wrapper object implementing a universal interface having a plurality of virtual functions each indexed by a number corresponding to an index number of a method associated with an interface of said requested COM object, and providing said client with a reference pointer to said wrapper COM object.

**EVIDENCE APPENDIX**

None

**RELATED PROCEEDINGS APPENDIX**

None